

LAPORAN RESMI
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
INHERITANCE 1



Fadilah Fahrul Hardiansyah S.ST., M. Kom

Ratri Maria Manik

3121600039

D4 TEKNIK INFORMATIKA – B

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
TA 2022/2023

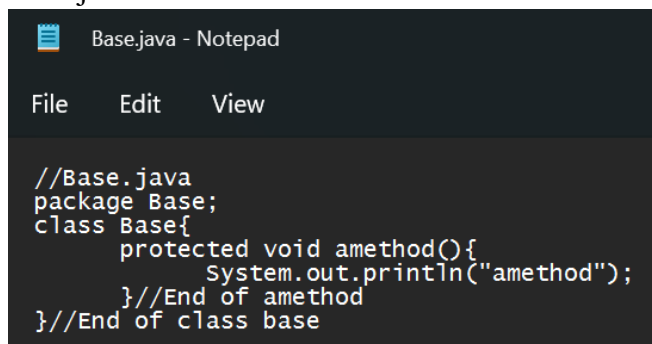
A. TUGAS PENDAHULUAN

1. Apa yang dimaksud inheritance?
Inheritance adalah sebuah konsep pewarisan sifat berupa variabel dan fungsi yang dimiliki oleh class untuk diwariskan kepada kelas – kelas yang lain.
2. Buatlah contoh kasus yang menerapkan konsep inheritance!
Setiap kasus yang memiliki beberapa class dengan atribut atau method yang sama antar class – class tersebut, maka dapat menggunakan konsep inheritance. Misalnya terdapat dua buah kelas yang memiliki atribut yang sama yaitu nama dan alamat.
3. Adakah perbedaan cara mengakses member class milik parent dan member class milik sendiri? Jelaskan melalui contoh! (manfaatkan jawaban soal nomor 2)
Adapun perbedaan cara pengaksesan antara member class milik parent dan member class milik sendiri yaitu, apabila ingin mengakses member class milik parent dapat menggunakan “super.data_member” sedangkan jika pengaksesan dilakukan terhadap child dapat menggunakan “this.data_member”, dan jika pengaksesan dilakukan terhadap parameter dapat langsung menuliskan parameter.
4. Apa yang dimaksud dengan konsep single inheritance?
Single inheritance adalah suatu subclass yang mempunyai satu parent class
5. Apa yang dimaksud dengan konsep multilevel inheritance?
Multilevel inheritance adalah suatu subclass mempunyai subclass lagi.

B. LATIHAN

1. Tempatkan base class dan class Class1 di direktori yang sama. Apa yang terjadi ketika class1.java dikompile dan dijalankan jika sebelumnya Base.java belum dikompile? Jelaskan!

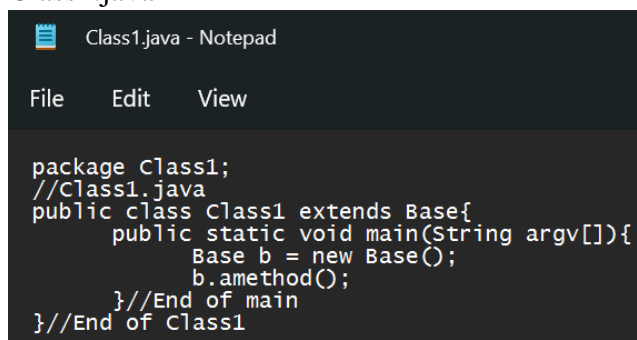
- Base.java



```
Base.java - Notepad
File Edit View

//Base.java
package Base;
class Base{
    protected void amethod(){
        System.out.println("amethod");
    }//End of amethod
}//End of class base
```

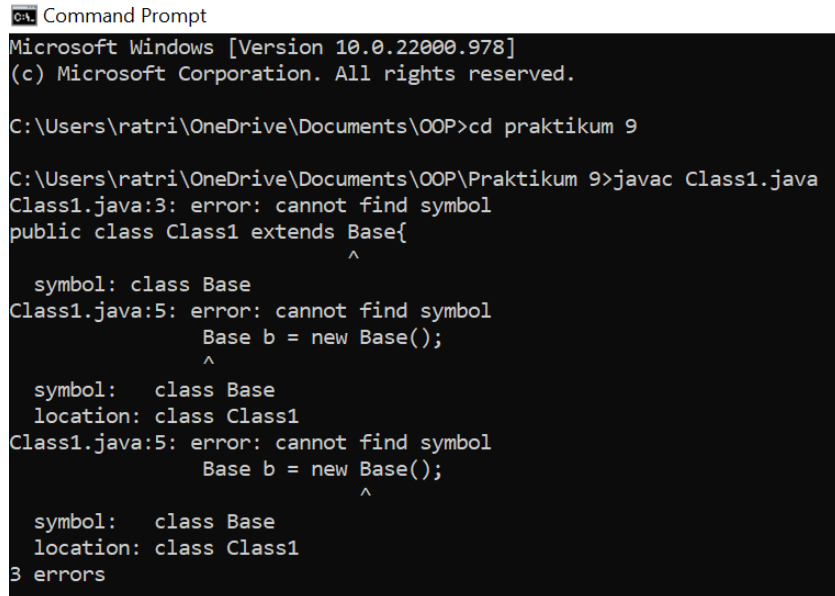
- Class1.java



```
Class1.java - Notepad
File Edit View

package Class1;
//Class1.java
public class Class1 extends Base{
    public static void main(String argv[]){
        Base b = new Base();
        b.amethod();
    }//End of main
}//End of Class1
```

- Hasil compile



```

CA. Command Prompt
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ratri\OneDrive\Documents\OOP>cd praktikum 9

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>javac Class1.java
Class1.java:3: error: cannot find symbol
public class Class1 extends Base{
                        ^
    symbol: class Base
Class1.java:5: error: cannot find symbol
    Base b = new Base();
                ^
    symbol:   class Base
    location: class Class1
Class1.java:5: error: cannot find symbol
    Base b = new Base();
                ^
    symbol:   class Base
    location: class Class1
3 errors

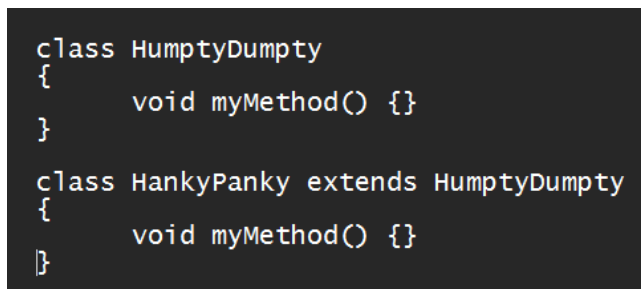
```

Analisa :

Terjadi error, sebaiknya modifier pada class Base.java diubah terlebih dahulu menjadi public, lalu menyertakan import package letak class Base.java, mengubah modifier method amethod() di dalam class Base.java menjadi public.

2. Aturan overriding

- Berdasarkan kode di bawah ini, akses modifier (public, protected atau private) apa yang diijinkan di tambahkan sebelum myMethod() baris 3?
- Jika baris 3 seperti kode di bawah (apa adanya tanpa perubahan) keywords apa yang diijinkan ditambahkan sebelum myMethod baris 8?



```

class HumptyDumpty
{
    void myMethod() {}
}

class HankyPanky extends HumptyDumpty
{
    void myMethod() {}
}

```

Analisa :

- Yang diizinkan hanya modifier private, karena akses modifier di subclassnya default, maka akses modifier yang diizinkan hanya private.
- Jika pada baris ke 3 diasumsikan default maka modifier yang diperbolehkan protected dan public
 - Jika overridden method yaitu default maka overriding harus default, protected atau public.
 - Jika overridden method yaitu protected maka overriding harus protected atau public.
 - Jika overridden method yaitu public, maka overriding harus public.

3.

- a. Apa yang terjadi bila kedua kode dibawah ini dikompile dan dijalankan dalam satu direktori? Jelaskan !

- P1.java

```
P1.java - Notepad
File Edit View

//File P1.java
package MyPackage;

class P1{
    void aFancyMethod(){
        System.out.println("What a fancy method");
    }
}
```

- P2.java

```
P2.java - Notepad
File Edit View

//File P2.java
public class P2 extends P1{
    public static void main(String argv[]){
        P2 p2 = new P2();
        p2.aFancyMethod();
    }
}
```

- Hasil kompilasi

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>javac P2.java
P2.java:2: error: cannot access P1
public class P2 extends P1{
      ^
bad source file: .\P1.java
file does not contain class P1
Please remove or make sure it appears in the correct subdirectory of the sourcepath.
P2.java:5: error: cannot find symbol
        p2.aFancyMethod();
           ^
symbol:   method aFancyMethod()
location: variable p2 of type P2
2 errors
```

- b. Bagaimana solusi supaya tidak terjadi error?

Melakukan import package terhadap class P1.java disimpan, lalu mengubah modifier class P1 menjadi public agar dapat diakses oleh class manapun, serta mengubah modifier method aFancyMethod menjadi public.

- P1.java

```
P1.java - Notepad
File Edit View

//File P1.java
package MyPackage;

public class P1{
    public void aFancyMethod(){
        System.out.println("What a fancy method");
    }
}
```

- Hasil compile

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>javac P2.java
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>java P2.java
What a fancy method
```

Analisa :

Error yang terjadi disebabkan karena tidak dilakukan import pada class P2.java, serta pada class P1.java belum dilakukan penyimpanan dipackage MyPackage. Selanjutnya pada program class P1.java modifier class diubah menjadi public agar dapat diakses oleh class yang lain begitu juga pada bagian methodnya.

4. Mengimplementasikan UML class diagram dalam program untuk package perbankan

- Class Tabungan.java

```
package perbankan;

public class Tabungan{
    protected int saldo;
}
```

- Class PenyimpananUang.java

```
package perbankan;

public class PenyimpananUang extends Tabungan{
    private double tingkatBunga;

    public PenyimpananUang(int saldo, double tingkatBunga){
        this.saldo = saldo;
        this.tingkatBunga = tingkatBunga;
    }

    public double cekUang(){
        return(this.saldo * tingkatBunga) + this.saldo;
    }
}
```

- Class TesLatihan.java

```
import perbankan.*;

public class TesLatihan{
    public static void main(String args[]){
        PenyimpananUang tabungan = new PenyimpananUang(5000,
        8.5/100);
        System.out.println("Uang yang ditabung : 5000");
        System.out.println("Tingkat bunga sekarang : 8.5%");
        System.out.println("Total uang anda sekarang : " +
        tabungan.cekUang());
    }
}
```

- Hasil compile

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>javac TesLatihan.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>java TesLatihan.java
Uang yang ditabung : 5000
Tingkat bunga sekarang : 8.5%
Total uang anda sekarang : 5425.0
```

Analisa :

Membuat suatu package perbankan yang menyimpan class Tabungan dan class PenyimpananUang, kemudian mengubah modifier pada class - class tersebut, dan juga modifier pada method yang dimiliki oleh class tersebut menjadi modifier public, selanjutnya dapat dilihat pada variabel saldo yang berada pada class Tabungan memiliki modifier protected.

C. TUGAS

1. Mengimplementasikan UML class diagram dalam program untuk package perbankan

- Class PengambilanUang.java

```
package perbankan;

public class PengambilanUang extends Tabungan{
    private int proteksi;

    public PengambilanUang(int saldo){
        this.saldo = saldo;
    }

    public PengambilanUang(int saldo, int tingkatBunga){
        this.saldo = saldo;
        this.proteksi = tingkatBunga;
    }

    public int getSaldo(){
        return saldo;
    }

    public boolean ambilUang(int jumlah){
        if(jumlah > saldo - proteksi){
            return false;
        }
        else{
            saldo -= jumlah;
            return true;
        }
    }
}
```

- Class TesTugas.java

```
import perbankan.*;

public class TesTugas{
    public static void main(String args[]){
```

```

        PengambilanUang tabungan = new PengambilanUang(5000, 1000);
        System.out.println("Uang yang ditabung : 5000");
        System.out.println("Uang yang diproteksi : 1000");
        System.out.println("-----");
        System.out.println("Uang yang akan diambil : 4500 " +
tabungan.ambilUang(4500));
        System.out.println("Saldo sekarang : " + tabungan.getSaldo());
        System.out.println("-----");
        System.out.println("Uang yang akan diambil : 2500 " +
tabungan.ambilUang(2500));
        System.out.println("Saldo sekarang : " + tabungan.getSaldo());
    }
}

```

- Hasil kompilasi

```

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>javac TestTugas.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 9>java TestTugas.java
Uang yang ditabung : 5000
Uang yang diproteksi : 1000
-----
Uang yang akan diambil : 4500 false
Saldo sekarang : 5000
-----
Uang yang akan diambil : 2500 true
Saldo sekarang : 2500

```

Analisa :

Dengan menggunakan package yang sama pada soal sebelumnya, hanya saja menambahkan suatu class yaitu PengambilanUang, yang memiliki atribut proteksi dengan modifier private.